Fast Multidimensional Asymptotic and Approximate Consensus

Matthias Függer, Thomas Nowak

Fast Multidimensional Asymptotic and Approximate Consensus

Matthias Függer
CNRS, LSV, ENS Paris-Saclay, Université Paris-Saclay, Inria
mfuegger@lsv.fr

Thomas Nowak
Université Paris-Sud
thomas.nowak@lri.fr

Abstract

We study the problems of asymptotic and approximate consensus in which agents have to get their values arbitrarily close to each others' inside the convex hull of initial values, either without or with an explicit decision by the agents. In particular, we are concerned with the case of multidimensional data, i.e., the agents' values are d-dimensional vectors. We introduce two new algorithms for dynamic networks, subsuming classical failure models like asynchronous message passing systems with Byzantine agents. The algorithms are the first to have a contraction rate and time complexity independent of the dimension d. In particular, we improve the time complexity from the previously fastest approximate consensus algorithm in asynchronous message passing systems with Byzantine faults by Mendes et al. [Distrib. Comput. 28] from $\Omega\big(d \log \frac{d\Delta}{\varepsilon}\big)$ to $O\big(\log \frac{\Delta}{\varepsilon}\big)$, where Δ is the initial and ε is the terminal diameter of the set of vectors of correct agents.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases asymptotic consensus; approximate consensus; multidimensional data; dynamic networks; Byzantine processes

Digital Object Identifier 10.4230/LIPIcs.DISC.2018.27

Funding This research was partially supported by the CNRS project PEPS DEMO and the Institut Farman.

1 Introduction

The problem of one-dimensional asymptotic consensus requires a system of agents, starting from potentially different initial real values, to repeatedly set their local output variables such that all outputs converge to a common value within the convex hull of the inputs. This problem has been studied in distributed control theory both from a theoretical perspective [10, 19, 5, 2] and in the context of robot gathering on a line [3] and clock synchronization [20, 16]. Extensions of the problem to multidimensional values naturally arise in the context of robot gathering on a plane or three-dimensional space [11], as subroutines in formation forming [10], and distributed optimization [4], among others.

The related problem of approximate consensus, also called approximate agreement, requires the agents to eventually decide, i.e., to only set their output variables once. Additionally all output variables must be within a predefined $\varepsilon > 0$ distance of each other and lie within the convex hull of the inputs. There is a large body of work on approximate consensus in distributed computing devoted to solvability and optimality of time complexity [13, 14] and applications in clock synchronization; see e.g. [24, 23].

Both problems were studied under different assumptions on the underlying communication between agents and their computational strength, including fully connected asynchronous message passing with Byzantine agents [24, 13] and communication in rounds by message passing in dynamic communication networks [19, 10]. In [6, 7] Charron-Bost et al. analyzed solvability of asymptotic consensus and approximate consensus in dynamic networks with round-wise message passing defined by *network models*: a network model is a set of directed communication graphs, each of which specifies successful reception of broadcast messages; see Section 2.1 for a formal definition. Solving asymptotic consensus in such a model requires to fulfill the specification of asymptotic consensus in *any* sequence of communication graphs from the model. Charron-Bost et al. showed that in these highly dynamic networks, asymptotic consensus and approximate consensus are solvable in a network model if and only if each of its graphs contains a spanning rooted tree. An interesting class of network models are those that contain only *non-split* communication graphs, i.e., communication graphs where each pair of nodes has a common incoming neighbor. Several classical fault-models were shown to be instances of non-split models [6], among them asynchronous message passing systems with omissions.

Recently the multidimensional version of approximate consensus received attention. Mendes et al. [18] were the first to present algorithms that solve approximate consensus in Byzantine message passing systems for d-dimensional real vectors. Their algorithms, Mendes–Herlihy and Vaidya–Garg, are based on the repeated construction of so called safe areas of received vectors to constraint influence of values sent by Byzantine agents, followed by an update step, ensuring that the new output values are in the safe area. They showed that the diameter of output values contracts by at least $1/2$ in each dimension every d rounds in the Mendes–Herlihy algorithm, and the diameter of the output values contracts by at least $1 - 1/n$ every round in the Vaidya–Garg algorithm, where n is the number of agents. The latter bound assumes $f = 0$ Byzantine failures and slightly worsens for $f > 0$. In terms of contraction rates as introduced in [15] (see Section 2.3 for a definition) of the respective non-terminating algorithms for asymptotic consensus, they thus obtain upper bounds of $\sqrt[d]{1/2}$ and $1 - 1/n$. Note that the Mendes–Herlihy algorithm has a contraction rate depending only on d but requires an a priori common coordinate system, and the algorithm's outcome depends on the choice of this coordinate system. By contrast the Vaidya–Garg algorithm is coordinate-free, i.e., its outcome is invariant under coordinate transformations such as translation and rotation, but it has a contraction rate depending on n.

Charron-Bost et al. [8] analyzed convergence of the Centroid algorithm where agents repeatedly update their position to the centroid of the convex hull of received vectors. The algorithm is coordinate-free and has a contraction rate of $d/(d+1)$, independent of n. Local time complexity of determining the centroid was shown to be #P-complete [21] while polynomial in n for fixed d.

The contraction rate of the Centroid algorithm is always smaller or equal to that of the Mendes-Herlihy algorithm, though both contraction rates converge to 1 at the same speed with the dimension d going to infinity. More precisely,

$$\lim_{d\to\infty} \frac{\left|1 - \sqrt[d]{\frac{1}{2}}\right|}{\left|1 - \frac{d}{d+1}\right|} = \log 2 \ ,$$

which implies $\left|1 - \sqrt[d]{\frac{1}{2}}\right| = \Theta\left(\left|1 - \frac{d}{d+1}\right|\right)$.

	MidExtremes	ApproachExtreme	Centroid	MH	VG
contraction rate	$\sqrt{\frac{7}{8}}^*$	$\sqrt{\frac{31}{32}}^*$	$\frac{d}{d+1}$	$\sqrt[d]{\frac{1}{2}}$	$1-\frac{1}{n}$
local TIME	$O(n^2 d)$	$O(nd)$	#P-hard	$O(nd)$	$O(nd)$
coordinate-free	yes	yes	yes	no	yes

Table 1 Comparison of local time complexity and contraction rates in non-split network models. Entries marked with an * are new results in this paper.

1.1 Contribution

In this work we present two new algorithms that are coordinate-free: the MidExtremes and the ApproachExtreme algorithm, and study their behavior in dynamic networks. Both algorithms are coordinate-free, operate in rounds, and are shown to solve asymptotic agreement in non-split network models. Terminating variants of them are shown to solve approximate agreement in non-split network models.

As a main result we prove that their contraction rate is independent of network size n and dimension d of the initial values. For MidExtremes we obtain an upper bound on the contraction rate of $\sqrt{7/8}$ and for ApproachExtreme of $\sqrt{31/32}$.

Due to the fact that classical failure models like asynchronous message passing with Byzantine agents possess corresponding network models, our results directly yield improved algorithms for the latter failure models: In particular, we improve the time complexity from the previously fastest approximate consensus algorithm in asynchronous message passing systems with Byzantine faults, the Mendes–Herlihy algorithm, from $\Omega\big(d \log \frac{d\Delta}{\varepsilon}\big)$ to $O\big(\log \frac{\Delta}{\varepsilon}\big)$, where Δ is the initial and ε is the terminal diameter of the set of vectors of correct agents. Note that our algorithms share the benefit of being coordinate-free with the Vaidya–Garg algorithm presented in the same work.

Table 1 summarizes our results and the algorithms discussed above for asymptotic and approximate consensus. The table compares the new algorithms MidExtremes and ApproachExtreme to the Centroid, Mendes–Herlihy (MH), and Vaidya–Garg (VG) algorithms with respect to their local time complexity per agent and round and an upper bound on their contraction rate in non-split network models. A lower bound of $1/2$ on the contraction rate is due to Függer et al. [15].

The Mendes-Herlihy algorithm has a smaller contraction rate than the MidExtremes algorithm whenever $d \leqslant 10$; the Centroid algorithm whenever $d \leqslant 14$. The Centroid algorithm is hence the currently fastest known algorithm for dimensions $3 \leqslant d \leqslant 14$. For dimensions $d = 1$ and $d = 2$, the componentwise MidPoint algorithm has an optimal contraction rate of $1/2$ [8]. Note that the MidExtremes algorithm is equivalent to the componentwise MidPoint algorithm for dimension $d = 1$. For $d \geqslant 15$, the MidExtremes algorithm is the currently fastest known algorithm.

We finally note that all our results hold for the class of inner product spaces and are not restricted to the finite-dimensional Euclidean spaces $\mathbb{R}^d$, in contrast to previous work. For example, this includes the set of square-integrable functions on a real interval. However, finite value representation and means to calculate the norm have to be guaranteed. Further, local TIME becomes n^2, respectively, n norm calculations.

2 Model and Problem

We fix some vector space V with an inner product $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ and the norm $\|x\| = \sqrt{\langle x, x \rangle}$. The prototypical finite-dimensional example is $V = \mathbb{R}^d$ with the usual inner product and the Euclidean norm. The diameter of set $A \subseteq V$ is denoted by $\operatorname{diam}(A) = \sup_{x,y \in A} \|x - y\|$. For an n-tuple $x = (x_1, \ldots, x_n) \in V^n$ of vectors in V, we write $\operatorname{diam}(x)$ by slight abuse of notation to denote $\operatorname{diam}\big(\{x_1, \ldots, x_n\}\big)$.

2.1 Dynamic Network Model

We consider a distributed system of n agents that communicate in rounds via message passing, like in the Heard-Of model [9]. In each round, each agent i, broadcasts a message based on its local state, receives some messages, and then updates its local state based on the received messages and its local state. Rounds are communication closed: agents only receive messages sent in the same round.

In each round $t \geqslant 0$, messages are delivered according to the *directed communication graph* G_t for round t: the message broadcast by i in round t is received by j if and only if the directed edge (i, j) is in G_t. Agents always receive their own messages, i.e., $(i, i) \in G_t$. A *communication pattern* is an infinite sequence $G_1, G_2, \ldots$ of communication graphs. A (deterministic) *algorithm* specifies, for each agent i, the local state space of i, the set of initial states of i, the sending function for which message to broadcast, and the state transition function. For asymptotic consensus, each agent i's local state necessarily contains a variable $y_i \in V$, which initially holds i's input value and is then used as its output variable. We require that there is an initial state with initial value v for all vectors $v \in V$. A *configuration* is an n-tuple of local states. It is called initial if all local states are initial. The *execution* of an algorithm from initial configuration C_0 induced by communication pattern $G_1, G_2, \ldots$ is the unique sequence $C_0, G_1, C_1, G_2, C_2, \ldots$ alternating between configurations and communication graphs where C_t is the configuration obtained by delivering messages in round t according to communication graph G_t, and applying the sending and local transition functions to the local states in C_{t-1} according to the algorithm. For a fixed execution and a local variable z of the algorithm, we denote by $z_i(t)$ its value at i at the end of round t, i.e., in configuration C_t. In particular, $y_i(t)$ is the value of y_i in C_t. We write $y(t) = \big(y_1(t), \ldots, y_n(t)\big)$ for the collection of the $y_i(t)$.

A specific class of algorithms for asymptotic consensus are the so-called *convex combination*, or *averaging*, algorithm, which only ever update the value of y_i inside the convex hull of y_j it received from other agents j in the current round. Many algorithms in the literature belong to this class, as do ours.

Following [6], we study the behavior of algorithms for communication patterns from a *network model*, i.e., a non-empty set of communication patterns: a communication pattern is from network model $\mathcal{N}$ if all its communication graphs are in $\mathcal{N}$. We will later on show that such an analysis also allows to prove new performance bounds for more classical fault-models like asynchronous message passing systems with Byzantine agents.

An interesting class of network models are so called *non-split* models, i.e., those that contain only non-split communication graphs: a communication graph is non-split if every pair of nodes has a common in-neighbor. Charron-Bost et al. [6] showed that asymptotic and approximate consensus is solvable efficiently in these network models in the case of one dimensional values. They further showed that: (i) In the weakest (i.e., largest) network model in which asymptotic and approximate consensus are solvable, the network model of

all communication graphs that contain a rooted spanning tree, one can simulate non-split communication graphs. (ii) Classical failure models like link failures as considered in [22] and asynchronous message passing systems with crash failures have non-split interpretations. Indeed we will make use of such a reduction from non-split network models to asynchronous message passing systems with Byzantine failures in Section 3.2.

2.2 Problem Formulation

An algorithm *solves the asymptotic consensus problem* in a network model $\mathcal{N}$ if the following holds for every execution with a communication pattern from $\mathcal{N}$:

- *Convergence.* For every agent i, the sequence $\big(y_i(t)\big)_{t\geqslant 0}$ converges.
- *Agreement.* If $y_i(t)$ and $y_j(t)$ converge, then they have a common limit.
- *Validity.* If $y_i(t)$ converges, then its limit is in the convex hull of the initial values $y_1(0),\dots,y_n(0)$.

For the deciding version, the *approximate consensus* problem (see, e.g., [17]), we augment the local state of i with a variable d_i initialized to $\perp$. Agent i is allowed to set d_i to some value $v\neq\perp$ only once, in which case we say that i *decides* v. In addition to the initial values $y_i(0)$, agents initially receive the error tolerance ε and an upper bound Δ on the maximum distance of initial values. An algorithm *solves approximate consensus* in $\mathcal{N}$ if for all $\varepsilon>0$ and all Δ, each execution with a communication pattern in $\mathcal{N}$ with initial diameter at most Δ satisfies:

- *Termination.* Each agent eventually decides.
- *ε-Agreement.* If agents i and j decide d_i and d_j, respectively, then $\|d_i-d_j\|\leqslant\varepsilon$.
- *Validity.* If agent i decides d_i, then d_i is in the convex hull of initial values $y_1(0),\dots,y_n(0)$.

2.3 Performance Metrics

A direct natural performance metric to assess the speed of convergence of agent outputs y along an execution is the *round-by-round convergence rate*

$$c(t)=\frac{\operatorname{diam}\big(y(t)\big)}{\operatorname{diam}\big(y(t-1)\big)}$$

for a given round $t\geqslant 1$ in the respective execution. The round-by-round convergence rate is the supremum over all executions and rounds. While a uniform upper bound of $\beta<1$ on the round-by-round convergence rate establishes convergence of the outputs, this measure fails in establishing convergence and comparing speeds of convergence for several algorithms considered in literature that set their output values every $k>1$ rounds, or that do not converge during an initial phase.

The *convergence rate*, defined by

$$\limsup_{t\to\infty}\sqrt[t]{\operatorname{diam}\big(y(t)\big)}\ ,$$

allows a comparison in this case by measuring eventual amortized convergence speed. For example, an algorithm that eventually contracts by a factor $\beta<1$ every $k\geqslant 1$ rounds has a convergence rate of $\sqrt[k]{\beta}$.

As a performance measure for general asymptotic consensus algorithms, where agents do not necessarily set their outputs y to within the convex hull of previously received vales, [15] considered the *contraction rate*, measuring contraction of reachable output limits rather

than output values: Following [15], the *valency* of a configuration C, denoted by $Y^*(C)$, is defined as the set of limits of the values y_i in executions that include configuration C. If the execution is clear from the context, we abbreviate $Y^*(t) = Y^*(C_t)$. The *contraction rate* of an execution is then defined as

$$\limsup_{t\to\infty} \sqrt[t]{\operatorname{diam}\left(Y^*(t)\right)} \ .$$

The contraction rate of an algorithm in a network model is the supremum of the contraction rates of its executions. For convex combination algorithms, the contraction rate is always upper-bounded by its convergence rate, that is,

$$\limsup_{t\to\infty} \sqrt[t]{\operatorname{diam}\left(Y^*(t)\right)} \leqslant \limsup_{t\to\infty} \sqrt[t]{\operatorname{diam}\left(y(t)\right)} \ ,$$

since the set of reachable limits $Y^*(t)$ at round t is contained in the set of output values $\{y_1(t), \dots, y_n(t)\}$ at round t for these algorithms.

Clearly, an algorithm that guarantees a round-by-round convergence rate of $c(t) \leqslant \beta$ also guarantees a convergence rate of at most β. Since both of our algorithms are convex combination algorithms, all our upper bounds on the round-by-round convergence rates are also upper bounds for the contraction rates.

The *convergence time* of a given execution measures the time from which on all values are guaranteed to be in an ε of each other. Formally, it is the function defined as

$$T(\varepsilon) = \min\left\{t \geqslant 0 \mid \forall \tau \geqslant t\colon \operatorname{diam}\left(y(\tau)\right) \leqslant \varepsilon\right\} \ .$$

In an execution that satisfies $c(t) \leqslant \beta$ for all $t \geqslant 1$, we have the bound $T(\varepsilon) \leqslant \left\lceil \log_{1/\beta} \frac{\Delta}{\varepsilon} \right\rceil$ on the convergence time, where $\Delta = \operatorname{diam}\left(y(0)\right)$ is the diameter of the set of initial values.

3 Algorithms

In this section, we introduce two new algorithms for solving asymptotic and approximate consensus in arbitrary inner product spaces with constant contraction rates. We present our algorithms and prove their correctness and bounds on their performance in non-split networks models. While we believe that this framework is the one in which our arguments are clearest, our results can be extended to a number of other models whose underlying communication graphs turn out to be, in fact, non-split. The following is a selection of these models:

- Rooted network models: This is the largest class of network models in which asymptotic and approximate consensus are solvable [6]. A network model is rooted if all its communication graphs include a directed rooted spanning tree, though not necessarily the same in all graphs. Although not every such communication graph is non-split, Charron-Bost et al. [6] showed that the cumulative communication graph over $n-1$ rounds in a rooted network model is always non-split. In such network models, one can use amortized versions [7] of the algorithms, which operate in macro-rounds of $n-1$ rounds each. If an algorithm has a contraction rate β in non-split network models, then its amortized version has contraction rate $\sqrt[n-1]{\beta}$ in rooted network models. The amortized versions of our algorithms thus have contraction rates independent of the dimension of the data.
- Omission faults: In the omission fault model studied by Santoro and Widmayer [22], the adversary can delete up to t messages from a fully connected communication graph each

round. If $t \leqslant n-1$, then all communication graphs are non-split. If $t \leqslant 2n-3$, then all communication graphs are rooted [6]. Our algorithms are hence applicable in both these cases and have contraction rates independent of the dimension.

- Asynchronous message passing with crash faults: Building asynchronous rounds atop of asynchronous message passing by waiting for $n-f$ messages in each round, the resulting communication graphs are non-split as long as the number f of possible crashes is strictly smaller than $n/2$. We hence get a constant contraction rate using our algorithms also in this model. For $f \geqslant n/2$, a partition argument shows that neither asymptotic nor approximate consensus are solvable.
- Asynchronous message passing with Byzantine faults: Mendes et al. [18] showed that approximate consensus is solvable in asynchronous message passing systems with f Byzantine faults if and only if $n > (d+2)f$ where d is the dimension of the data. The algorithms they presented construct a round structure whose communication graphs turn out to be non-split. Since the construction is not straightforward, we postpone the discussion of our algorithms in this model to Section 3.2.

3.1 Non-split Network Models

We now present our two new algorithms, MidExtremes and ApproachExtreme. Both operate in the following simple round structure: broadcast the current value y_i and then update it to a new value depending on the set Rcv_i of values y_j received from agents j in the current round. Both of them only need to calculate distances between values and form the midpoint between two values. In particular, we do not need to make any assumption on the dimension of the space of possible values for implementing the algorithms. We only need a distance and an affine structure, for calculating the midpoint. Our correctness proofs, however, rely on the fact that the distance function is a norm induced by an inner product.

Note that, although we present algorithms for asymptotic consensus, combined with our upper bounds on the convergence time, one can easily deduce versions for approximate consensus by having the agents decide after the upper bound. Our upper bounds only depend on the precision parameter ε and (an upper bound on) the initial diameter Δ. While upper bounds on the initial diameter cannot be deduced during execution in general non-split network models, it can be done in specific models, like asynchronous message passing with Byzantine faults [18]. Otherwise, we need to assume an a priori known bound on the initial diameter to solve approximate consensus.

The algorithm MidExtremes, which is shown in Algorithm 1, updates its value y_i to the midpoint of a pair of extremal points of Rcv_i that realizes its diameter. In the worst case, it thus has to compare the distances of $\Theta(n^2)$ pairs of values. For the specific case of Euclidean spaces $V = \mathbb{R}^d$ stored in a component-wise representation, this amounts to $O(n^2 d)$ local scalar operations for each agent in each round.

It turns out that we can show a round-by-round convergence rate of the MidExtremes algorithm independent of the dimension or the number of agents, namely $\sqrt{7/8}$. For the specific case of values from the real line $V = \mathbb{R}$, it reduces to the MidPoint algorithm [7], whose contraction rate of $1/2$ is known to be optimal [15].

THEOREM 1. *In any non-split network model with values from any inner product space, the MidExtremes algorithm guarantees a round-by-round convergence rate of $c(t) \leqslant \sqrt{7/8}$ for all rounds $t \geqslant 1$. Its convergence time is at most $T(\varepsilon) = \left\lceil \log_{\sqrt{8/7}} \frac{\Delta}{\varepsilon} \right\rceil$ where Δ is the diameter of the set of initial values.*

Algorithm 1 Asymptotic consensus algorithm MidExtremes for agent i

Initialization:

y_i is the initial value in V

In round $t \geqslant 1$ **do:**

broadcast y_i

$\mathrm{Rcv}_i \leftarrow$ set of received values

$(a, b) \leftarrow \underset{(a,b) \in Rcv_i^2}{\arg\max} \|a - b\|$

$y_i \leftarrow \dfrac{a+b}{2}$

In the particular case of values from the real line, it guarantees a round-by-round convergence rate of $c(t) \leqslant 1/2$ *and a convergence time of* $T(\varepsilon) = \left\lceil \log_2 \frac{\Delta}{\varepsilon} \right\rceil$.

The second algorithm we present is called ApproachExtreme and shown in Algorithm 2. It updates its value y_i to the midpoint of the current value of y_i and the value in Rcv_i that is the farthest from it. While having the benefit of only having to compare $O(n)$ distances, and hence doing $O(nd)$ local scalar operations for each agent in each round in the case of $V = \mathbb{R}^d$ with component-wise representation, the ApproachExtreme algorithm also only has to measure distances from its current value to other agents' values; never the distance of two other agents' values. This can be helpful for agents embedded into the vector space V that can measure the distance from itself to another agent, but not necessarily the distance between two other agents.

Algorithm 2 Asymptotic consensus algorithm ApproachExtreme for agent i

Initialization:

y_i is the initial value in V

In round $t \geqslant 1$ **do:**

broadcast y_i

$\mathrm{Rcv}_i \leftarrow$ set of received values

$b \leftarrow \underset{b \in Rcv_i}{\arg\max} \|y_i - b\|$

$y_i \leftarrow \dfrac{y_i + b}{2}$

The ApproachExtreme algorithm admits an upper bound of $\sqrt{31/32}$ on its round-by-round convergence rate, which is worse than the $\sqrt{7/8}$ of the MidExtremes algorithm. For the case of the real line $V = \mathbb{R}$, we can show a round-by-round convergence rate of $3/4$, however.

THEOREM 2. *In any non-split network model with values from any inner product space, the ApproachExtreme algorithm guarantees a round-by-round convergence rate of* $c(t) \leqslant \sqrt{\frac{31}{32}}$ *for all rounds* $t \geqslant 1$. *Its convergence time is at most* $T(\varepsilon) = \left\lceil \log_{\sqrt{32/31}} \frac{\Delta}{\varepsilon} \right\rceil$ *where* Δ *is the diameter of the set of initial values.*

In the particular case of values from the real line, it guarantees a round-by-round convergence rate of $c(t) \leqslant 3/4$ *and a convergence time of* $T(\varepsilon) = \left\lceil \log_{4/3} \frac{\Delta}{\varepsilon} \right\rceil$.

3.2 Asynchronous Byzantine Message Passing

We now show how to adapt algorithm MidExtremes to the case of asynchronous message passing systems with at most f Byzantine agents. The algorithm proceeds in the same asynchronous round structure and safe area calculation used by Mendes et al. [18] whenever approximate consensus is solvable, i.e., when $n > (d+2)f$. Plugging in the MidExtremes algorithm, we achieve a round-by-round convergence rate and round complexity independent of the dimension d.

More specifically, our algorithm has a round complexity of $O\big(\log \frac{\Delta}{\varepsilon}\big)$, which leads to a message complexity of $O\big(n^2 \log \frac{\Delta}{\varepsilon}\big)$ where Δ is the maximum Euclidean distance of initial vectors of correct agents. In contrast, the Mendes-Herlihy algorithm has a worst-case round complexity of $\Omega\big(d \log \frac{d\Delta}{\varepsilon}\big)$ and a worst-case message complexity of $\Omega\big(n^2 d \log \frac{d\Delta}{\varepsilon}\big)$. We are thus able to get rid of all terms depending on the dimension d.

After an initial round estimating the initial diameter of the system, the Mendes-Herlihy algorithm has each agent i repeat the following steps in each coordinate $k \in \{1, 2, \ldots, d\}$ for $\Theta\big(\log \frac{d\Delta}{\varepsilon}\big)$ rounds:

1. Collect a multiset V_i of agents' vectors such that every intersection $V_i \cap V_j$ has at least $n - f$ elements via reliable broadcast and the witness technique [1].
2. Calculate the safe area S_i as the intersection of the convex hulls of all sub-multisets of V_i of size $|V_i| - f$. The safe area is guaranteed to be a subset of the convex hull of vectors of correct agents. Helly's theorem [12] can be used to show that every intersection $S_i \cap S_j$ of safe areas is nonempty.
3. Update the vector y_i to be in the safe area S_i and have its k^{th} coordinate equal to the midpoint of the set of k^{th} coordinates in S_i.

The fact that safe areas have nonempty pairwise intersections guarantees that the diameter in the k^{th} coordinate

$$\delta_k(t) = \max_{i,j \text{ correct}} \left| y_i^{(k)}(t) - y_j^{(k)}(t) \right|$$

at the end of round t fulfills $\delta_k(t) \leqslant \delta_k(t-1)/2$ if round t considers coordinate k. The choice of the number of rounds for each coordinate guarantees that we have $\delta_k(t) \leqslant \varepsilon/\sqrt{d}$ after the last round for coordinate k. This in turn makes sure that the Euclidean diameter of the set of vectors of correct agents after all of the $\Theta\big(d \log \frac{d\Delta}{\varepsilon}\big)$ rounds is at most ε.

The article of Mendes et al. [18] describes a second algorithm, the Vaidya-Garg algorithm, which replaces steps 2 and 3 by updating y_i to the non-weighted average of arbitrarily chosen points in the safe areas of all sub-multisets of V_i of size $n - f$. Another difference to the Mendes-Herlihy algorithm is that it repeats the steps not several times for every dimension, but for $\Theta\big(n^{f+1} \log \frac{d\Delta}{\varepsilon}\big)$ rounds in total. The Vaidya-Garg algorithm comes with the advantage of not having to do the calculations to find a midpoint for the k^{th} coordinate while remaining inside the safe area, but also comes with the cost of a convergence rate and a round complexity that depends on the number of agents.

The algorithm we propose has the same structure as the Mendes-Herlihy algorithm, with the following differences: (i) like the Vaidya-Garg algorithm it is missing the loop over all coordinates one-by-one, and (ii) we replace step 3 by updating vector y_i to the midpoint of two points that realize the Euclidean diameter of the safe area S_i. According to our results in Section 4.1, the Euclidean diameter

$$\delta(t) = \max_{i,j \text{ correct}} \big\| y_i(t) - y_j(t) \big\|$$

of the set of vectors of correct agents at the end of round t satisfies

$$\delta(t) \leqslant \sqrt{\frac{7}{8}}\delta(t-1) \ .$$

This means that we have $\delta(T) \leqslant \varepsilon$ after

$$T(\varepsilon) = \left\lceil \log_{\sqrt{8/7}} \frac{\Delta}{\varepsilon} \right\rceil$$

rounds.

4 Performance Bounds

We next show upper bounds on the round-by-round convergence rate for algorithms MidExtremes (Theorem 1) and ApproachExtreme (Theorem 2) in non-split network models.

4.1 Bounds for MidExtremes

For dimension 1, MidExtremes is equivalent to the MidPoint Algorithm. We hence already know that $c(t) \leqslant \frac{1}{2}$ from [7], proving the case of the real line in Theorem 1.

For the case of higher dimensions we will show that $c(t) \leqslant \sqrt{\frac{7}{8}}$ holds. The proof idea is as follows: For a round $t \geqslant 1$, we consider two agents i, j whose distance realizes $\operatorname{diam}(y(t))$. By the algorithm we know that both agents set their $y_i(t)$ and $y_j(t)$ according to $y_i(t) = m = (a+b)/2$ and $y_j(t) = m' = (a'+b')/2$, where a, b are the extreme points received by agents i in round t and a', b' are the extreme points received by agents j in the same round. All four points must lie within a common subspace of dimension 3, and form the vertices of a tetrahedron as depicted in Figure 1.

Further, any three points among a, b, a', b' must lie within a 2 dimensional subspace, forming a triangle. Lemma 3 states the distance from the midpoint of two of its vertices to the opposite vertex, say c, and an upper bound in case the two edges incident to c are upper bounded in length.

Lemma 3. *Let $\gamma \geqslant 0$ and $a, b, c \in V$. Setting $m = (a+b)/2$, we have*

$$\|m-c\|^2 = \frac{1}{2}\|a-c\|^2 + \frac{1}{2}\|b-c\|^2 - \frac{1}{4}\|a-b\|^2 \ .$$

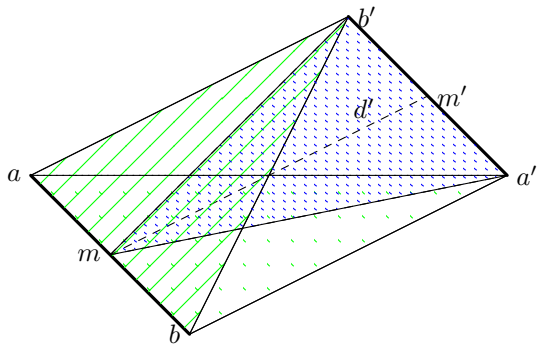

Figure 1 Tetrahedron formed by extreme points a and b of agent i and extreme points a' and b' of agent j. The distance between the new agent positions m and m' is d'.

In particular, if $\|a-c\| \leqslant \gamma$ *and* $\|b-c\| \leqslant \gamma$*, then*

$$\|m-c\|^2 \leqslant \gamma^2 - \frac{1}{4}\|a-b\|^2 \ .$$

Proof. We begin by calculating

$$\|a-c\|^2 = \big\|(a-m)+(m-c)\big\|^2 = \|a-m\|^2 + \|m-c\|^2 + 2\langle a-m, m-c\rangle \tag{1}$$

and

$$\|b-c\|^2 = \big\|(b-m)+(m-c)\big\|^2 = \|b-m\|^2 + \|m-c\|^2 + 2\langle b-m, m-c\rangle \ . \tag{2}$$

Adding (1) and (2), while noting $\|a-m\|^2 = \|b-m\|^2 = \frac{1}{4}\|a-b\|^2$ and $a-m = (a-b)/2 = -(b-m)$, gives

$$\|a-c\|^2 + \|b-c\|^2 = \frac{1}{2}\|a-b\|^2 + 2\|m-c\|^2 \ .$$

Rearranging the terms in the last equation concludes the proof. ◀

We are now in the position to prove Lemma 4 that is central for Theorem 1. The lemma provides an upper bound on the distance d' between m and m' for the tetrahedron in Figure 1 given that all its sides are upper bounded by some $\gamma \geqslant 0$ and the sum of the lengths of edge a, b and a', b', i.e., $\|a-b\| + \|a'-b'\|$, is lower bounded by γ. At the heart of the proof of Lemma 4 is an application of Lemma 3 for the three hatched triangles in Figure 1.

Lemma 4. *Let* $a, b, a', b' \in V$ *and* $\gamma \geqslant 0$ *such that*

$$\operatorname{diam}\big(\{a,b,a',b'\}\big) \leqslant \gamma \leqslant \|a-b\| + \|a'-b'\| \ . \tag{3}$$

Then, setting $m = (a+b)/2$ *and* $m' = (a'+b')/2$*, we have*

$$\|m-m'\| \leqslant \sqrt{\frac{7}{8}}\gamma \ .$$

Proof. Applying Lemma 3 with the points a, b, a' yields

$$\|m-a'\|^2 \leqslant \gamma^2 - \frac{1}{4}\|a-b\|^2 \ . \tag{4}$$

Another invocation with the points a, b, b' gives

$$\|m-b'\|^2 \leqslant \gamma^2 - \frac{1}{4}\|a-b\|^2 \ . \tag{5}$$

Now, again using Lemma 3 with the points a', b', m and the bounds of (4) and (5), we get

$$\|m-m'\|^2 \leqslant \gamma^2 - \frac{1}{4}\left(\|a-b\|^2 + \|a'-b'\|^2\right) \ .$$

Using the second inequality in (3) then shows

$$\|m-m'\|^2 \leqslant \gamma^2 - \frac{1}{4}\left(\|a-b\|^2 + \big(\gamma - \|a-b\|\big)^2\right) \ . \tag{6}$$

Setting $\xi = \|a-b\|$, we get

$$\|m-m'\|^2 \leqslant \max_{0\leqslant\xi\leqslant\gamma} \gamma^2 - \frac{1}{4}\big(\xi^2 + (\gamma-\xi)^2\big) \ .$$

Differentiating the function $f(\xi) = \gamma^2 - \frac{1}{4}\big(\xi^2 + (\gamma - \xi)^2\big)$ reveals that its maximum is attained for $-(2\xi - \gamma) = 0$, i.e., $\xi = \gamma/2$, which gives

$$\|m - m'\|^2 \leqslant \gamma^2 - \frac{\gamma^2}{8} = \frac{7}{8}\gamma^2 \ .$$

Taking the square root now concludes the proof. ◀

We can now prove Theorem 1. For the proof we consider the tetrahedron with vertices a, b, a', b' as discussed before; see Figure 1. Recalling that the vertices a, b are vectors received by an agent i and a', b' vectors received by an agent j in the same round, we may infer from the non-split property that all communication graphs must fulfill that both i and j must have received a common vector from an agent. Together with the algorithm's rule of picking a, b and a', b' as extreme points, we obtain the constraints required by Lemma 4. Invoking this lemma we finally obtain an upper bound on the distance d' between m and m', and by this an upper bound on the round-by-round convergence rate of the MidExtremes algorithm.

Proof of Theorem 1. Let i and j be two agents. Let $a, b \in \mathrm{Rcv}_i(t)$ such that $y_i(t) = (a+b)/2$ and $a', b' \in \mathrm{Rcv}_j(t)$ such that $y_j(t) = (a' + b')/2$. Define $\gamma_{ij} = \operatorname{diam}\big(\{a, b, a', b'\}\big)$. Since a, b, a', b' are the vectors of some agents in round $t-1$, we have $\gamma_{ij} \leqslant \operatorname{diam}\big(y(t-1)\big)$.

Further, from the non-split property, there is an agent k whose vector $c = y_k(t-1)$ has been received by both i and j, i.e., $c \in \mathrm{Rcv}_i(t) \cap \mathrm{Rcv}_j(t)$. By the choice of the extreme points a, b by agent i, we must have $\|a - c\| \leqslant \|a - b\|$; otherwise a, b would not realize the diameter of $\mathrm{Rcv}_i(t)$. Analogously, by the choice of the extreme points a', b' by agent j, it must hold that $\|a' - c\| \leqslant \|a' - b'\|$.

From the triangular inequality, we then obtain

$$\|a - a'\| \leqslant \|a - c\| + \|c - a'\| \leqslant \|a - b\| + \|a' - b'\| \ .$$

Analogous arguments for the other pairs of points in $\{a, b, a', b'\}$ yield

$$\operatorname{diam}\big(\{a, b, a', b'\}\big) = \gamma_{ij} \leqslant \|a - b\| + \|a' - b'\| \ .$$

We can hence apply Lemma 4 to obtain

$$\|y_i(t) - y_j(t)\| \leqslant \sqrt{\frac{7}{8}}\gamma_{ij} \leqslant \sqrt{\frac{7}{8}} \operatorname{diam}\big(y(t-1)\big) \ .$$

Taking the maximum over all pairs of agents i and j now shows $\operatorname{diam}\big(y(t)\big) \leqslant \sqrt{7/8} \operatorname{diam}\big(y(t-1)\big)$, which concludes the proof. ◀

4.2 Bounds for ApproachExtreme

We start by showing the one-dimensional case of Theorem 2, i.e., $V = \mathbb{R}$, in Section 4.2.1. Section 4.2.2 then covers the multidimensional case.

4.2.1 One-dimensional Case

For the proof we use the notion of ϱ-safety as introduced by Charron-Bost et al. [7]. A convex combination algorithm is *ϱ-safe* if

$$\varrho M_i(t) + (1 - \varrho) m_i(t) \leqslant y_i(t) \leqslant (1 - \varrho) M_i(t) + \varrho m_i(t) \tag{7}$$

where $M_i(t) = \max\big(\mathrm{Rcv}_i(t)\big)$ and $m_i(t) = \min\big(\mathrm{Rcv}_i(t)\big)$.

It was shown [7, Theorem 4] that any ϱ-safe convex combination algorithm guarantees a round-by-round convergence rate of $c(t) \leqslant 1-\varrho$ in any non-split network model. In the sequel, we will show that ApproachExtreme is $\frac{1}{4}$-safe when applied in $V = \mathbb{R}$.

Proof of Theorem 2, one-dimensional case. Let i be an agent and $t \geqslant 1$ a round in some execution of ApproachExtreme in $V = \mathbb{R}$. We distinguish the two cases $y_i(t) \leqslant y_i(t-1)$ and $y_i(t) > y_i(t-1)$.

In the first case, we have $b \leqslant y_i(t-1)$ for the vector b that agent i calculates in code line 4 in round t. But then necessarily $b = y_i(t)$ since this is the most distant point to $y_i(t-1)$ in $\mathrm{Rcv}_i(t)$ to the left of $y_i(t-1)$. Also, $y_i(t-1) \geqslant \big(M_i(t)+m_i(t)\big)/2$ since otherwise $M_i(t)$ would be farther from $y_i(t-1)$ than $m_i(t)$. But this means that

$$y_i(t) = \frac{y_i(t-1)+m_i(t)}{2} \geqslant \frac{1}{4}M_i(t) + \frac{1}{4}m_i(t) + \frac{1}{2}m_i(t) = \frac{1}{4}M_i(t) + \frac{3}{4}m_i(t) \ ,$$

which shows the first inequality of ϱ-safety (7) with $\varrho = \frac{1}{4}$. The second inequality of (7) follows from $y_i(t-1) \leqslant M_i(t)$ since

$$y_i(t) = \frac{y_i(t-1)+m_i(t)}{2} \leqslant \frac{1}{2}M_i(t) + \frac{1}{2}m_i(t) \leqslant \frac{3}{4}M_i(t) + \frac{1}{4}m_i(t) \ .$$

In the second case, (7) is proved analogously to the first case. ◀

4.2.2 Multidimensional Case

For the proof of Theorem 2 with higher dimensional values, we consider two agents i, j whose distance realizes $\mathrm{diam}(y(t))$. From the ApproachExtreme $y_i(t) = m = (a + y_i(t-1))/2$ and $y_j(t) = m' = (a' + y_j(t-1))/2$ where a and a' maximize the distance to $y_i(t-1)$ and $y_j(t-1)$, respectively, among the received values.

To show an upper bound on the distance d' between the new agent positions m and m' in the multidimensional case, we need the following variant of Lemma 4 in which we relax the upper bound on γ by a factor of two, but thereby weaken the bound on d'.

Analogous to the proof of Theorem 1, the proof is by applying Lemma 5 to the three hatched triangles in Figure 1.

LEMMA 5. *Let $a, b, a', b' \in V$ and $\gamma \geqslant 0$ such that*

$$\mathrm{diam}\big(\{a, b, a', b'\}\big) \leqslant \gamma \leqslant 2\|a-b\| + 2\|a'-b'\| \ .$$

Then, setting $m = (a+b)/2$ and $m' = (a'+b')/2$, we have

$$\|m - m'\| \leqslant \sqrt{\frac{31}{32}}\gamma \ .$$

The proof of the lemma is essentially the same as that of Lemma 4, with the following differences: Equation (6) is replaced by

$$\|m-m'\|^2 \leqslant \gamma^2 - \frac{1}{4}\left(\|a-b\|^2 + \left(\frac{\gamma}{2} - \|a-b\|\right)^2\right) \ ,$$

which changes the function f to $f(\xi) = \gamma^2 - \frac{1}{4}\big(\xi^2 + (\frac{\gamma}{2} - \xi)^2\big)$. The maximum of this function f is achieved for $\xi = \gamma/4$, which means that

$$\|m-m'\|^2 \leqslant f(\gamma/4) = \gamma^2 - \frac{\gamma^2}{32} = \frac{31}{32}\gamma^2 \ .$$

We are now in the position to prove Theorem 2.

Proof of Theorem 2, multidimensional case. Let i and j be two agents. Let $a = y_i(t-1)$ and $a' = y_j(t-1)$. Further, let $b \in \mathrm{Rcv}_i(t)$ such that $y_i(t) = (a+b)/2$ and $b' \in \mathrm{Rcv}_j(t)$ such that $y_j(t) = (a'+b')/2$. Define $\gamma_{ij} = \operatorname{diam}\big(\{a, b, a', b'\}\big)$. Since a, b, a', b' are the vectors of some agents in round $t-1$, we have $\gamma_{ij} \leqslant \operatorname{diam}\big(y(t-1)\big)$.

From the non-split property, there is an agent k whose vector $c = y_k(t-1)$ has been received by both i and j, i.e., $c \in \mathrm{Rcv}_i(t) \cap \mathrm{Rcv}_j(t)$. By the choice of the extreme point b by agent i, we must have $\|a-c\| \leqslant \|a-b\|$; otherwise b would not maximize the distance to a. Analogously, by the choice of the extreme points b' by agent j, it must hold that $\|a'-c\| \leqslant \|a'-b'\|$. Note, however, that the roles of a and b are not symmetric and that, contrary to the proof of Theorem 1, we can have $\|b-c\| > \|a-b\|$ or $\|b'-c\| > \|a'-b'\|$.

From the triangular inequality and the two established inequalities, we then obtain

$$\|a-a'\| \leqslant \|a-c\| + \|a'-c\| \leqslant \|a-b\| + \|a'-b'\| \ ,$$

$$\|a-b'\| \leqslant \|a-c\| + \|c-a'\| + \|a'-b'\| \leqslant \|a-b\| + 2\|a'-b'\| \ ,$$

and

$$\|b-b'\| \leqslant \|b-a\| + \|a-c\| + \|c-a'\| + \|a'-b'\| \leqslant 2\|a-b\| + 2\|a'-b'\| \ .$$

Analogously, $\|a'-b\| \leqslant 2\|a-b\| + \|a'-b'\|$. Together this implies

$$\operatorname{diam}\big(\{a, b, a', b'\}\big) = \gamma_{ij} \leqslant 2\|a-b\| + 2\|a'-b'\| \ .$$

We can hence apply Lemma 5 to obtain

$$\|y_i(t) - y_j(t)\| \leqslant \sqrt{\frac{31}{32}}\gamma_{ij} \leqslant \sqrt{\frac{31}{32}} \operatorname{diam}\big(y(t-1)\big) \ .$$

Taking the maximum over all pairs of agents i and j now shows $\operatorname{diam}\big(y(t)\big) \leqslant \sqrt{31/32} \cdot \operatorname{diam}\big(y(t-1)\big)$, which concludes the proof. ◀

5 Conclusion

We presented two new algorithms for asymptotic and approximate consensus with values in arbitrary inner product spaces. This includes not only the Euclidean spaces $\mathbb{R}^d$, but also spaces of infinite dimension. Our algorithms are the first to have constant contraction rates, independent of the dimension and the number of agents.

We have presented our algorithms in the framework of non-split network models and have then shown how to apply them in several other distributed computing models. In particular, we improved the round complexity of the algorithms by Mendes et al. [18] for asynchronous message passing with Byzantine faults from $\Omega\big(d \log \frac{d\Delta}{\varepsilon}\big)$ to $O\big(\log \frac{\Delta}{\varepsilon}\big)$, eliminating all terms that depend on the dimension d.

The exact value of the optimal contraction rate for asymptotic and approximate consensus is known to be $1/2$ in dimensions one and two [15, 8], but the question is still open for higher dimensions. Our results are a step towards the solution of the problem as they show the optimum in all dimensions to lie between $1/2$ and $\sqrt{7/8} \approx 0.9354\ldots$

References

1 Ittai Abraham, Yonatan Amit, and Danny Dolev. Optimal resilience asynchronous approximate agreement. In Teruo Higashino, editor, *8th International Conference on Principles of Distributed Systems (OPODIS 2004)*, volume 3544 of *Lecture Notes in Computer Science*, pages 229–239. Springer, Heidelberg, 2005.

2 David Angeli and Pierre-Alexandre Bliman. Stability of leaderless discrete-time multi-agent systems. *MCSS*, 18(4):293–322, 2006.

3 Zohir Bouzid, Maria Gradinariu Potop-Butucaru, and Sébastien Tixeuil. Optimal Byzantine-resilient convergence in uni-dimensional robot networks. *Theoretical Computer Science*, 411(34-36):3154–3168, 2010.

4 Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

5 Ming Cao, A. Stephen Morse, and Brian D. O. Anderson. Reaching a consensus in a dynamically changing environment: convergence rates, measurement delays, and asynchronous events. *SIAM J. Control Optim.*, 47(2):601–623, 2008.

6 Bernadette Charron-Bost, Matthias Függer, and Thomas Nowak. Approximate consensus in highly dynamic networks: The role of averaging algorithms. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming, ICALP15*, pages 528–539, 2015.

7 Bernadette Charron-Bost, Matthias Függer, and Thomas Nowak. Fast, robust, quantizable approximate consensus. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming, ICALP16*, pages 137:1–137:14, 2016.

8 Bernadette Charron-Bost, Matthias Függer, and Thomas Nowak. Multidimensional asymptotic consensus in dynamic networks. *CoRR*, abs/1611.02496, 2016. URL: `http://arxiv.org/abs/1611.02496`.

9 Bernadette Charron-Bost and André Schiper. The Heard-Of model: computing in distributed systems with benign faults. *Distrib. Comput.*, 22(1):49–71, 2009.

10 Bernard Chazelle. The total s-energy of a multiagent system. *SIAM Journal on Control and Optimization*, 49(4):1680–1706, 2011.

11 Mark Cieliebak, Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Solving the robots gathering problem. In *International Colloquium on Automata, Languages, and Programming*, pages 1181–1196. Springer, 2003.

12 Ludwig Danzer, Branko Grünbaum, and Victor Klee. Helly's theorem and its relatives. In Victor Klee, editor, *Convexity*, volume 7 of *Proceedings of Symposia in Pure Mathematics*, pages 101–180. AMS, Providence, 1963.

13 Danny Dolev, Nancy A. Lynch, Shlomit S. Pinter, Eugene W. Stark, and William E. Weihl. Reaching approximate agreement in the presence of faults. *Journal of the ACM*, 33(2):499–516, 1986.

14 Alan D. Fekete. Asymptotically optimal algorithms for approximate agreement. *Distrib. Comput.*, 4(1):9–29, 1990.

15 Matthias Függer, Thomas Nowak, and Manfred Schwarz. Tight bounds for asymptotic and approximate consensus. In *Symposium on Principles of Distributed Computing (PODC)*, 2018. To appear, preprint available at `https://arxiv.org/abs/1705.02898`.

16 Qun Li and Daniela Rus. Global clock synchronization in sensor networks. *IEEE Transactions on Computers*, 55(2):214–226, 2006.

17 Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, San Francisco, CA, 1996.

18 Hammurabi Mendes, Maurice Herlihy, Nitin Vaidya, and Vijay K. Garg. Multidimensional agreement in Byzantine systems. *Distributed Computing*, 28:423–441, 2015.

19 Luc Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.

20 Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9):1520–1533, 2004.

21 Luis A. Rademacher. Approximating the centroid is hard. In *Proceedings of the Twenty-third Annual Symposium on Computational Geometry*, pages 302–305. ACM, 2007.

22 Nicola Santoro and Peter Widmayer. Time is not a healer. In B. Monien and R. Cori, editors, *6th Symposium on Theoretical Aspects of Computer Science*, volume 349 of *LNCS*, pages 304–313. Springer, Heidelberg, 1989.

23 Fred B Schneider. Understanding protocols for Byzantine clock synchronization. Technical report, Cornell University, 1987.

24 Jennifer Lundelius Welch and Nancy Lynch. A new fault-tolerant algorithm for clock synchronization. *Information and computation*, 77(1):1–36, 1988.